ИНФОРМАТИКА, ВЫЧИСЛИТЕЛЬНАЯ ТЕХНИКА И УПРАВЛЕНИЕ
INFORMATION TECHNOLOGY, COMPUTER SCIENCE, AND MANAGEMENT



УДК 003.26 10.23947/1992-5980-2017-17-3-145-155

Алгоритмическая оценка сложности системы кодирования и защиты информации, основанной на пороговом разделении секрета, на примере системы электронного голосования*

Л. В. Черкесова[1], О. А. Сафарьян[2], А. В. Мазуренко[3], Н. С. Архангельская[4]**
[1, 2, 3, 4] Донской государственный технический университет, г. Ростов-на-Дону, Российская Федерация

Complexity calculation of coding and information security system based on threshold secret sharing scheme used for electronic voting***

L. V. Cherkesova[1], O. A. Safaryan[2], A. V. Mazurenko[3], N. S. Arkhangelskaya[4]**
[1, 2, 3, 4] Don State Technical University, Rostov-on-Don, Russian Federation

Введение. Одной из задач криптографии является обеспечение безопасного и честного проведения электронного голосования. При такой процедуре избиратели подают голоса в электронном виде — например, через электронные терминалы. В работе предложен новый алгоритм порогового разделения секрета для проведения электронного голосования.
Материалы и методы. При решении поставленной исследовательской задачи использованы теория конечных полей, теория алгоритмов, проективная геометрия и линейная алгебра. Разработанная криптосистема основана на применении геометрических объектов из проективной геометрии, что позволяет задействовать аппарат линейной алгебры для эффективного решения криптографических задач. Для оценки сложности работы описанных алгоритмов использованы классические результаты из теории алгоритмов.
Результаты исследования. В работе описаны криптографические алгоритмы разделения секрета и его последующего восстановления, основанные на использовании особенностей построения проективных пространств над конечными полями и их связи с полями Галуа подходящего порядка. Подробно описаны составные части данных алгоритмов, а именно: метод построения инъективного отображения, действующего из кольца вычетов по простому модулю в проективное пространство над конечным полем определенной размерности; способ генерации секретных долей и секрета; процедура разделение секрета и его последующего восстановления. Приведены алгоритмические оценки временной сложности описанных формальных алгоритмов.
Обсуждение и заключения. Предложенная схема может быть применена для проведения электронных выборов, а также в иных областях, где естественным образом возникает необходимость в применении методов пороговой криптографии.

Introduction. One of the tasks arising in cryptography is to ensure the safe and honest conduct of e-voting. This procedure provides that voters submit their votes electronically – for example, through electronic terminals. A new algorithm for the distribution of threshold sensitive data for electronic voting is proposed.
Materials and Methods. The results are obtained on the basis of the following methodology: finite field theory, theory of algorithms, projective geometry, and linear algebra. The developed cryptosystem is based on the application of geometric objects from projective geometry which makes it possible to use the apparatus of linear algebra to make effective decisions on cryptographic problems. To estimate the complexity of the described algorithms, classical results from the theory of algorithms are applied.
Research Results. This paper describes the cryptographic algorithms of secret sharing and its subsequent restoration based on special structural properties of projective spaces over finite fields, and their link with Galois fields of the appropriate order. The component parts of these algorithms, specifically, the construction of injective mapping from a residue ring prime modulo into the projective space over finite field of specific dimension; the generation of secret shares and secret; the procedure of secret sharing and its restoration, are described in great detail. The algorithmic time complexity calculations of the formal algorithms are given.
Discussion and Conclusions. The described scheme is useful for electronic voting and in other spheres where methods of threshold cryptography are applied.

Ключевые слова: криптография, электронное голосование, пороговая криптография, разделение секрета, криптосистема Эль-Гамаля, криптосистема с открытым ключом, криптографический секрет, криптографический алгоритм, информационная безопасность, криптографический ключ

Keywords: cryptography, electronic voting, threshold cryptography, secret sharing, ElGamal encryption system, public-key cryptography, cryptographic secret, cryptographic algorithm, information security, cryptographic key.

*Работа выполнена в рамках инициативной НИР.
**E-mail: mazurencoal@gmail.com, arh.iv@bk.ru, chia2002@inbox.ru, safari_2006@mail.ru
*** The research is done within the frame of the independent R&D.

Образец для цитирования: Черкесова, Л. В. Алгоритмическая оценка сложности системы кодирования и защиты информации, основанной на пороговом разделении секрета, на примере системы электронного голосовани / Л. В. Черкесова, О. А. Сафарьян, А. В. Мазуренко, Н. С. Архангельская // Вестник Дон. гос. техн. ун-та. — 2017. — Т.17, № 3. — С. 145‒155.

For citation: I L.V. Cherkesova, O.A. Safaryan, A.V. Mazurenko, N.S. Arkhangelskaya. Complexity calculation of coding and information security system based on threshold secret sharing scheme used for electronic voting. Vestnik of DSTU, 2017, vol. 17, no.3, pp. 145‒155.

Введение. Одной из задач, которые возникают в криптографии, является обеспечение безопасного и честного проведения электронного голосования. Такая процедура предусматривает, что избиратели подают голоса в электронном виде — например, через электронные терминалы. Это ключевое отличие электронного голосования от традиционных выборов, в ходе которых избиратели заполняют бюллетени на избирательных участках, а подсчет голосов осуществляется избирательными комиссиями вручную. До недавнего времени электронное голосование рассматривалось как теоретическая задача в области криптографии. Однако в последние несколько лет положение дел изменилось. В США и в странах Европейского Союза активно обсуждаются возможности проведения президентских выборов и выборов в парламенты этих государств с использованием электронных систем.

В созданной ранее криптосистеме [1] описан алгоритм порогового разделения секрета для проведения электронного голосования. В криптосистеме [1] можно выделить три вида участников: проверяющие, избиратели и администратор. Администратор генерирует закрытый и публичный ключи согласно усиленному варианту схемы Эль-Гамаля [2, 3]. Закрытый ключ разбивается на секретные доли, которые распределяются между проверяющими, состоящими в одной команде. Для каждой команды проверяющих создаются свои уникальные секретные доли. Далее избиратели голосуют за одну из кандидатур и зашифровывают свой «голос» при помощи открытого ключа. Зашифрованное сообщение отправляется на сервер, доступ к которому имеют только проверяющие. После завершения голосования команды проверяющих распределяют между собой шифротексты, восстанавливают секретный ключ и дешифруют принятые от избирателей сообщения. В результате все команды предоставляют дешифрованные голоса, суммируют их и объявляют результаты выборов. Сложность взлома построенной криптосистемы эквивалента сложности решения общепризнанно трудной задачи принятий решения Диффи-Хеллмана в некоторой циклической Абелевой группе [4–7].

Алгоритмы порогового разделения секрета встречаются, например, в [8].

Актуальность данной статьи определяется необходимостью оценить сложность построенной криптосистемы в [1], что, в свою очередь, позволит выяснить, является ли данная система потенциально привлекательной для использования на практике.

Постановка задачи. Авторы ставят задачу описания формальных алгоритмов, необходимых для реализации метода разделения секрета, представленного в [1]. Кроме того, будет дана временная оценка сложности описанных алгоритмов.

Основная часть. Предположим, администратор сгенерировал некоторый секретный ключ. На основе секрета ему необходимо создать секретные доли, которые будут распределены между участниками. Поскольку секрет является элементом некоторого кольца вычетов по простому модулю, то сначала необходимо построить инъективное отображение из этого кольца в проективное пространство над конечным полем определенной размерности. Сопоставив секрету точку проективного пространства, администратор строит некоторую проективную прямую, проходящую через эту точку. Данная прямая публикуется. Затем администратор строит проективное подпространство, которое пересекает известную прямую ровно в одной точке, причем это та самая точка, которая соответствует секрету. Размерность построенного подпространства равна количеству участников в команде проверяющих минус один. Далее в рассматриваемом проективном подпространстве выделяется система проективно независимых точек, причем их количество совпадает с количеством проверяющих, и они порождают данное подпространство. Именно эти проективно независимые точки служат долями секрета. Для восстановления секрета проверяющие должны построить проективное подпространство, найдя прямую сумму своих секретных долей, и пересечь его с известной проективной прямой. Итак, найдена точка, соответствующая секрету. При помощи левого обратного отображения к построенной инъекции находится сам секрет.

Корректность нижеследующих алгоритмов обсуждается в статье [1]. Приведенные результаты описывают генерацию секретных долей для одной команды проверяющих. Последующее восстановление секретного ключа описано также только для одной команды. Для остальных команд проводятся аналогичные действия.

Обозначим N множество натуральных чисел. Далее везде $1 < t$, $t \in \mathrm{N}$ — число проверяющих, входящих в одну команду, $p < w^{m+1}$, где p — простое число, w — положительная степень некоторого простого числа, $m \in \mathrm{N}$: $t \le m$.

Обозначим $PG(m,w)$ проективное пространство над конечным полем F_w размерности m. Все точки $PG(m,w)$ представимы в виде $(\beta^i : \delta\beta^i : \ldots : \delta^{w-2}\beta^i)$, где $i \in \overline{0, n-1}$, $n = \dfrac{w^{m+1}-1}{w-1}$, δ — примитивный элемент поля F_w, β — примитивный элемент поля $F_{w^{m+1}}$ [1].

Обозначим VS^k структуру данных, которая хранит некоторым образом упорядоченные элементы векторного пространства F_w^k над полем F_w, $k \in \mathrm{N}$. Для данной функции g обозначение $O(g)$ означает множество функций — таких, что

$$O(g) = \{ f : \exists (C > 0), n_0 : \forall (n > n_0)\ 0 \le f(n) \le Cg(n) \},$$

где C, n_0 — положительные константы, O-нотация используется для оценки временной сложности алгоритмов [3].

Секрет, который разделяется между проверяющими, является случайно сгенерированным элементом мультипликативной группы кольца вычетов по простому модулю p, то есть $x \in \overline{1, p-1}$. Поэтому вначале построим инъективное отображение, сопоставляющее x некоторую точку $PG(m,w)$ конечной проективной геометрии над конечным полем F_w размерности m.

Как было установлено в [1], в состав публичного ключа входят: α — примитивный элемент поля Z_p, β — примитивный элемент поля $F_{w^{m+1}}$. Далее везде α и β несут такой смысл. Итак, можно полагать, что секретный ключ представим в виде $x = \alpha^i \in \mathrm{Z}_p^*$, где $i \in \overline{0, p-2}$.

Пусть $F_q = F_w[x]/(f)$ — произвольное конечное поле, $q = w^e$ — положительная степень некоторого простого числа w, $e \in N$, $f \in F_w[x]$ — примитивный многочлен. Каждый элемент F_q представим в виде полинома, принадлежащего $F_w[x]$, степень которого не превосходит $e \in N$. Тогда для сложения и вычитания двух элементов поля F_q используется $A(q) = O(\log_2 q)$ бит операций. В качестве алгоритма умножения в конечном поле F_q будем использовать алгоритм Карацубы [9]. Итак, временная сложность операции умножения двух полиномов из F_q равна $O\left((e \log_2 q)^{\log_2 3}\right)$. Результат умножения необходимо поделить на примитивный многочлен $f \in F_w[x]$. Временная сложность операции деления равна $O\left(e(e \log_2 q)^{\log_2 3}\right) = O\left(e^{1+\log_2 3} (\log_2 q)^{\log_2 3}\right)$. Итак, окончательно можно положить, что временная сложность умножения в конечном поле F_q равна $M(q) = M(w^e) = O\left(e^{1+\log_2 3} (\log_2 q)^{\log_2 3}\right)$. Для быстрого возведения в степень в конечном поле F_q используется схема «слева направо», описанная в [10], временная сложность которой равна

$$ME(q) = O\left(M(q) \log_2 q\right) = O\left((e^2 \log_2 w)^{1+\log_2 3}\right).$$

Сопоставим секрету $x \in Z_p^*$ некоторый элемент конечного поля — так, чтобы его можно было в дальнейшем восстановить. Следующий формальный алгоритм описывает один из возможных способов построить инъективное отображение $\mu : \mathrm{Z}_p^* \to F_{w^{m+1}}^*$. Найдем $\mu(x)$, где $x \in \mathrm{Z}_p^*$.

Алгоритм 1. Построение $\mu : \mathrm{Z}_p^* \to F_{w^{m+1}}^*$.

$FirstMapping\left(\beta : F_{w^{m+1}}^* = <\beta>, i \in \overline{0, p-2} : x = \alpha^i \in \mathrm{Z}_p^*\right)$

1. $\mu(x) \leftarrow \beta^i$;
2. *return* $\mu(x)$.

Лемма 1. Алгоритм 1 нахождения значения отображения μ имеет временную сложность $O\left(((m+1)^2 \log_2 w)^{1+\log_2 3}\right)$.

Далее сопоставим элементы конечного поля точкам проективного пространства. Найдем результат действия сюръекции $\tau : F_{w^{m+1}}^* \to PG(m,w)$, $\tau(y)$, где $y \in F_{w^{m+1}}^*$.

Алгоритм 2. Построение $\tau : F_{w^{m+1}}^* \to PG(m,w)$.

$SecondMapping\left(\beta : F_{w^{m+1}}^* = <\beta>, m, w, k \in \overline{0, w^{m+1}-2} : y = \beta^k \in F_{w^{m+1}}^*\right)$

1. $n \leftarrow \dfrac{w^{m+1}-1}{w-1}$;

2. $r \leftarrow k (\mathrm{mod}\, n)$;

3. $\tau(y) \leftarrow (\beta^r)$;

4. *return* $\tau(y)$.

Лемма 2. Алгоритм 2 нахождения значения отображения τ имеет временную сложность $O\left(((m+1)^2 \log_2 w)^{1+\log_2 3}\right)$.

Доказательство. Шаг 1 выполняется за время $O\left(\log_2(m+1)\cdot\left(\log_2 w^{m+1}\right)^{\log_2 3}\right)$, так как временная сложность двух вычитаний, одного деления и возведения в степень при работе с целыми числами равна

$$O\left(\log_2 w^{m+1} + \log_2 w + \left(\log_2 w^{m+1}\right)^{\log_2 3} + \log_2(m+1)\cdot\left(\log_2 w^{m+1}\right)^{\log_2 3}\right) =$$

$$= O\left(\log_2(m+1)\cdot\left(\log_2 w^{m+1}\right)^{\log_2 3}\right).$$

Шаг 2 выполняется за время $O\left(\left(\log_2 w^{m+1}\right)^{\log_2 3}\right)$, а шаг 3 — за время $ME(w^{m+1})$. Таким образом, получаем оценку

$$O\left(\log_2(m+1)\cdot\left(\log_2 w^{m+1}\right)^{\log_2 3}\right) + O\left(\left(\log_2 w^{m+1}\right)^{\log_2 3}\right) + O\left(((m+1)^2 \log_2 w)^{1+\log_2 3}\right) =$$

$$= O\left(((m+1)^2 \log_2 w)^{1+\log_2 3}\right) = ME(w^{m+1}).$$

Опишем инъективное отображение $\varphi : Z_p^* \to PG(m,w)\times F_w^*$, при помощи которого секретному ключу сопоставляется пара, состоящая из точки проективного пространства $PG(m,w)$ над полем F_w и элемента F_w^*, необходимого для построения левого обратного отображения к φ. Найдем $\varphi(x)$, где $x \in \mathrm{Z}_p^*$.

Алгоритм 3. Построение $\varphi : Z_p^* \to PG(m,w)\times F_w^*$.

$ThirdMapping\,(\beta : F_{w^{m+1}}^* = <\beta>, m, w, i \in \overline{0, p-2} : x = \alpha^i \in Z_p^*)$

1. $\tau\left(\mu(x)\right) \leftarrow SecondMapping\left(\beta, m, w, i\right)$; // $\mu(x) = \beta^i$;

2. $n \leftarrow \dfrac{w^{m+1}-1}{w-1}$;

3. $k \leftarrow i - i(\mathrm{mod}\, n)$;

4. вычислить $\beta^k \in F_{w^{m+1}}^*$;

5. $\varphi(x) \leftarrow \left(\tau\left(\mu(x)\right), \beta^k\right)$;

6. *return* $\varphi(x)$.

Лемма 3. Алгоритм 3 нахождения значения отображения φ имеет временную сложность $O\left(((m+1)^2 \log_2 w)^{1+\log_2 3}\right)$.

Доказательство. Из леммы 2 следует, что шаг 1 выполняется за время $ME(w^{m+1})$, шаг 2 — $O\left(\log_2(m+1)\cdot\left(\log_2 w^{m+1}\right)^{\log_2 3}\right)$. На шаге 3 выполняется одно деление и одно вычитание целых чисел; время выполнения данных операций равно $O\left(\left(\log_2 w^{m+1}\right)^{\log_2 3} + \log_2 w^{m+1}\right) = O\left(\left(\log_2 w^{m+1}\right)^{\log_2 3}\right)$. Шаг 4 выполняется за время $ME(w^{m+1})$. Шаг 5 выполняется за константное время. Итак, временная сложность алгоритма равна

$$ME(w^{m+1}) + O\left(\log_2(m+1)\cdot\left(\log_2 w^{m+1}\right)^{\log_2 3}\right) + O\left(\left(\log_2 w^{m+1}\right)^{\log_2 3}\right) = ME(w^{m+1}). \bullet$$

Далее возникает необходимость в генерации попарно различных случайных точек проективного пространства $PG(m,w)$ над полем F_w. Опишем один из способов сгенерировать $t \in N$ попарно различных случайных точек, где $t \le n$, $n = \frac{w^{m+1}-1}{w-1}$. Обозначим через $random([0,a])$, где $a \in N$, алгоритм, реализующий линейный конгруэнтный метод генерации псевдослучайных чисел [6], который подает на выход целое случайное число в промежутке $[0,a]$.

Алгоритм 4. Генерирование попарно различных случайных точек $PG(m,w)$.

$GeneratePoints(\beta : F^*_{w^{m+1}} = <\beta>, m, w, t)$

1. $n \leftarrow \frac{w^{m+1}-1}{w-1}$;
2. $Powers \leftarrow \varnothing$;
3. $for\ (i \leftarrow 0; i < t; i++)\ \{$
4. $do\ \{$
5. $temp \leftarrow random([0, n-1])$;
6. $\}\ while(temp \in Powers)$;
7. $Powers[i] \leftarrow temp$;
8. $\}$
9. $Points \leftarrow \varnothing$;
10. $for\ (i \leftarrow 0; i < t; i++)\ \{$
11. $Points[i] \leftarrow SecondMapping(\beta, m, w, Powers[i]);\ \}$
12. *return Points.*

Лемма 4. Алгоритм 4 генерирования попарно различных случайных точек $PG(m,w)$ имеет временную сложность $O(t^3)ME(w^{m+1})$.

Доказательство. Шаг 1 выполняется за время $O\left(\log_2(m+1)\cdot\left(\log_2 w^{m+1}\right)^{\log_2 3}\right)$. Шаг 2 выполняется за константное время. Шаг 5 выполняется за время $O(1)$, шаг 6 — $O(t)$, цикл в строках 4–6 — $O(t^2)$. Шаг 7 выполняется за константное время. Итак, цикл в строках 3–8 выполняется за $O(t^3)$. Шаг 9 выполняется за константное время. Цикл в строках 10–11 выполняется за время $O(t)ME(w^{m+1})$. Таким образом, временная сложность алгоритма равна

$$O\left(\log_2(m+1)\cdot\left(\log_2 w^{m+1}\right)^{\log_2 3}\right) + O(t^3) + O(t)ME(w^{m+1}) = O(t^3)ME(w^{m+1}).$$

Опишем построение правого обратного отображения к сюръекции τ: $\tau^{-1} : PG(m,w) \to F^*_{w^{m+1}}$. Найдем $\tau^{-1}((y))$, где $(y) \in PG(m,w)$. Здесь $n = \frac{w^{m+1}-1}{w-1}$.

Алгоритм 5. Построение $\tau^{-1} : PG(m,w) \to F^*_{w^{m+1}}$.

$InvSecMap\left(\beta : F^*_{w^{m+1}} = <\beta>, k \in \overline{0, n-1} : (y) = (\beta^k) \in PG(m,w)\right)$

1. $\tau^{-1}((y)) \leftarrow \beta^k$;
2. *return* $\tau^{-1}(x)$.

Лемма 5. Алгоритм 5 построения отображения τ^{-1} имеет вычислительную сложность $O\left(((m+1)^2 \log_2 w)^{1+\log_2 3}\right)$.

Доказательство. Шаг 1 выполняется за время $ME(w^{m+1})$.

Для восстановления секрета используется проективная прямая $l \subset PG(m,w)$, содержащая секретную точку. Данная прямая является частью публичного ключа. Опишем работу алгоритма 6 *LineConstruc-*

$tion\left(VS^2, \beta : F^*_{w^{m+1}} =<\beta>, m, w, (x) = (\beta^k) \in PG(m,w)\right)$ построения проективной прямой $l \subset PG(m,w)$, которая содержит заданную точку $(x) = (\beta^k) \in PG(m,w)$, $k \in \overline{0, n-1}$, $n = \dfrac{w^{m+1}-1}{w-1}$.

Алгоритм 6. Построение проективной прямой $l \subset PG(m,w)$.

$LineConstruction\left(VS^2, \beta : F^*_{w^{m+1}} =<\beta>, m, w, (x) = (\beta^k) \in PG(m,w)\right)$

1. *do* {
2. $(y) = (\beta^v) \leftarrow GeneratePoints(\beta, m, w, 1)$;
3. } *while* $\left((x) = (y)\right)$;
4. $l \leftarrow \varnothing$;
5. *for* $(i \leftarrow 0; i < w^2; i++)$ {
6. $temp \leftarrow VS^2[i][0] \cdot InvSecMap(\beta, k) + VS^2[i][1] \cdot InvSecMap(\beta, v)$;
7. $newPoint \leftarrow SecondMapping(\beta, m, w, z)$; //$temp = \beta^z$, $z \in \overline{0, w^{m+1}-2}$
8. $l \leftarrow l \cup \{\text{newPoint}\}$;
9. }
10. *return* l.

Лемма 6. Алгоритм 6 построения проективной прямой $l \subset PG(m,w)$ имеет временную сложность $O(w^3)ME(w^{m+1})$.

Опишем один из возможных способов создания проективного подпространства $M \subset PG(m,w)$ размерности $t-1 \in \overline{0, m}$, которое содержит произвольную, но зафиксированную точку $(x) = (\beta^k) \in PG(m,w)$, $k \in \overline{0, n-1}$, $n = \dfrac{w^{m+1}-1}{w-1}$. Умение строить такие проективные подпространства необходимо для генерации секретных долей.

Алгоритм 7. Построение проективного подпространства $M \subset PG(m,w)$ **размерности** $t-1$.

$SubspaceConstruction\left(VS^t, \beta : F^*_{w^{m+1}} =<\beta>, m, w, t, (x) = (\beta^k) \in PG(m,w)\right)$

1. $Points \leftarrow GeneratePoints(\beta, m, w, t)$;
2. *if* ($(x) \notin Points$);
3. $Points[0] \leftarrow (x)$;
4. $M \leftarrow \varnothing$;
5. *for* $(i \leftarrow 0; i < w^t; i++)$ {
6. $temp \leftarrow 0$;
7. *for* $(j \leftarrow 0; j < t; j++)$ {
8. $temp \leftarrow temp + VS^t[i][j] \cdot InvSecMap(\beta, Points[j])$; }
9. $M \leftarrow M \cup \{SecondMapping(\beta, m, w, z)\}$; //$temp = \beta^z$, $z \in \overline{0, w^{m+1}-2}$
10. }
11. *return M*.

Лемма 7. Алгоритм 7 построения проективного подпространства $M \subset PG(m,w)$ размерности $t-1$ имеет временную сложность $O(t^3 + tw^{2t})ME(w^{m+1})$.

Доказательство. Согласно лемме 4 шаг 1 выполняется за время $O(t^3)ME(w^{m+1})$. Цикл в строках 2–3 выполняется за время $O\left(\dfrac{w^t-1}{w-1}\right) = O(w^t)$. Шаги 4 и 6 выполняются за константное время. Согласно лемме 5 шаг 8 выполняется за время $ME(w^{m+1}) + M(w^{m+1}) + A(w^{m+1}) = ME(w^{m+1})$. Итак, цикл в строках 7–8 выполняется за время

$O(t)ME(w^{m+1})$. Согласно лемме 2 шаг 9 выполняется за время $O\left(\frac{w^t-1}{w-1}\right)+ME(w^{m+1})=O(w^t)+ME(w^{m+1})$, так как $M\subset PG(m,w)$ содержит $\frac{w^t-1}{w-1}$ точек. Таким образом, цикл в строках 5–10 выполняется за время

$$O(tw^t)ME(w^{m+1})+O(w^{2t})+O(w^t)ME(w^{m+1})=O(tw^{2t})ME(w^{m+1}).$$

Итак, временная сложность алгоритма равна

$$O(t^3)ME(w^{m+1})+O(w^t)+O(tw^{2t})ME(w^{m+1})=O(t^3+tw^{2t})ME(w^{m+1}).$$

Легко проверить, что количество проективных подпространств размерности $d\in\overline{0,m}$ в $PG(m,w)$ равно $[d]_w^m=\frac{\left(w^{m+1}-1\right)\left(w^m-1\right)\ldots\left(w^{m-d+1}-1\right)}{\left(w^{d+1}-1\right)\left(w^d-1\right)\ldots\left(w-1\right)}$.

Опишем способ построения проективного подпространства $M\subset PG(m,w)$ размерности $t-1\in\overline{0,m}$: $M\cap l=(x)$, где $l\subset PG(m,w)$ — проективная прямая, (x) — произвольная, но зафиксированная точка, принадлежащая l.

Легко проверить, что количество проективных подпространств размерности $d\in\overline{0,m}$ в $PG(m,w)$ равно

$$[d]_w^m=\frac{\left(w^{m+1}-1\right)\left(w^m-1\right)\ldots\left(w^{m-d+1}-1\right)}{\left(w^{d+1}-1\right)\left(w^d-1\right)\ldots\left(w-1\right)}.$$

Алгоритм 8. Построение проективного подпространства $M\subset PG(m,w)$ размерности $t-1$: $M\cap l=(x)$.

GenerateSpecialSubspace $(VS^t,\beta:F^*_{w^{m+1}}=<\beta>,m,w,t,l,(x)\in l)$

1. *while* (*true*) {
2. $M\leftarrow$ *SubspaceConstruction* $(VS^t,\beta,m,w,t,(x))$;
3. *intersection* $\leftarrow M\cap l$;
4. *if* (*intersection* $=\{(x)\}$);
5. *break*;
6. }
7. *return M*.

Легко увидеть, что верна лемма 8.

Лемма 8. Алгоритм 8 построения проективного подпространства $M\subset PG(m,w)$ размерности $t-1$: $M\cap l=(x)$, где $l\subset PG(m,w)$ — проективная прямая — такая, что $(x)\in l$, имеет временную сложность $O\left([t-1]_w^m(t^3+tw^{2t})\right)ME(w^{m+1})$.

Под проективным базисом проективного подпространства $M\subset PG(m,w)$ будем понимать проективно независимые точки M — такие, что их прямая сумма порождает *M*. Опишем один из способов построения проективного базиса проективного подпространства $M\subset PG(m,w)$ размерности $t-1\in\overline{0,m}$, который не содержит произвольную, но зафиксированную точку $(x)\in M$.

Алгоритм 9. Построение проективного базиса проективного подпространства $M\subset PG(m,w)$ размерности $t-1$.

GenerateSubspaceBasisSet $\left(w,t,M,(x)\in M\right)$

1. $|M|\leftarrow\frac{w^t-1}{w-1}$;
2. *do* {
3. $T\leftarrow$ *выбрать* t *попарно различных точек, принадлежащих M*;
4. } *while* $\left((x)\in T\right)$;
5. *return T*.

Легко увидеть, что верна лемма 9.

Лемма 9. Алгоритм 9 построения проективного базиса проективного подпространства $M \subset PG(m,w)$ размерности $t-1$ имеет временную сложность $O\left(\left(\log_2 w^t\right)^{\log_2 3} + \binom{w^t}{t}\right)$.

Опишем процедуру создания и разделения секретных долей среди $t \in \overline{2,m}$ проверяющих на основе секрета $x = \alpha^k \in Z_p^*$, $k \in \overline{0, p-2}$.

Алгоритм 10. Генерация и разделение секретных долей.

$GenerateAndShareSecrets\left(VS^2, VS^t, \beta : F^*_{w^{m+1}} =<\beta>, m, w, t, k : x = \alpha^k \in Z_p^*\right)$

1. $\varphi(x) = \left((y) = (\beta^i) \in PG(m,w), r \in F^*_{w^{m+1}}\right) \leftarrow ThirdMapping(\beta, m, w, k)$;
2. $l \leftarrow LineConstruction\left(VS^2, \beta, m, w, (y)\right)$;
3. $M \leftarrow GenerateSpecialSubspace\,(VS^t, \beta, m, w, t, l, (y))$;
4. $T \leftarrow GenerateSubspaceBasisSet\left(w, t, M, (y)\right)$;
5. *return T.*

Теорема 1. Алгоритм 10 генерации и раздачи секретных долей среди t проверяющих имеет временную сложность

$$O\left([t-1]_w^m (t^3 + tw^{2t})\right) ME(w^{m+1}) + O\left(\binom{w^t}{t}\right).$$

Доказательство. Результат следует из лемм 3, 6, 8 и 9, согласно которым временная сложность алгоритма равна

$$ME(w^{m+1}) + O(w^3) ME(w^{m+1}) + O\left([t-1]_w^m (t^3 + tw^{2t})\right) ME(w^{m+1}) +$$
$$+ O\left(\left(\log_2 w^t\right)^{\log_2 3} + \binom{w^t}{t}\right) = O\left([t-1]_w^m (t^3 + tw^{2t})\right) ME(w^{m+1}) + O\left(\binom{w^t}{t}\right).$$

Опишем формальный алгоритм восстановления проективного подпространства $M \subset PG(m,w)$ при наличии проективного базиса *T* этого подпространства.

Алгоритм 11. Построение проективного подпространства $M \subset PG(m,w)$ **размерности** $t-1$ **по проективному базису *T*** $M \subset PG(m,w)$.

$BasisSubspaceConstruction\left(VS^t, \beta : F^*_{w^{m+1}} =<\beta>, w, t, T\right)$

1. *for* $(i \leftarrow 0; i < w^t; i{+}{+})$ {
2. *temp* $\leftarrow 0$;
3. *for* $(j \leftarrow 0; j < t; j{+}{+})$ {
4. *temp* $\leftarrow$ *temp*+ $VS^t\,[i][j] \cdot InvSecMap\,(\beta, T[j])$; }
5. $M \leftarrow M \cup \{SecondMapping(\beta, m, w, z)\}$; //*temp*= β^z, $z \in \overline{0, w^{m+1}-2}$;
6. }
7. *return M.*

Легко увидеть, что верна лемма 10.

Лемма 10. Алгоритм 11 построения проективного подпространства $M \subset PG(m,w)$ размерности $t-1$ по базису *T* этого подпространства имеет временную сложность $O(tw^{2t}) \cdot ME(w^{m+1})$.

Опишем построение левого обратного отображения к инъекции μ: $\mu^{-1} : F^*_{w^{m+1}} \rightarrow Z_p^*$. Найдем $\mu^{-1}(y)$, где $y \in F^*_{w^{m+1}}$.

Алгоритм 12. Построение $\mu^{-1}: F^*_{w^{m+1}} \to Z^*_p$.

$InverseFirstMapping\left(\alpha : Z^*_p =<\alpha>, k \in \overline{0, p-2} : y = \beta^k \in F^*_{w^{m+1}}\right)$

1. $\mu^{-1}(x) \leftarrow \alpha^k$;

2. $return\ \mu^{-1}(x)$.

Лемма 11. Алгоритм 12 нахождения значения отображения μ^{-1} имеет временную сложность $O\left((\log_2 p)^{1+\log_2 3}\right)$.

Доказательство. Шаг 1 выполняется за время $ME(p)$.

Опишем построение левого обратного отображения к инъекции φ: $\varphi^{-1}: PG(m,w)\times F^*_w \to Z^*_p$. Найдем $\varphi^{-1}\left((y),s\right)$, где $(y)\in PG(m,w)$, $s \in F^*_w$.

Алгоритм 13. Построение $\varphi^{-1}: PG(m,w)\times F^*_w \to Z^*_p$.

$InverseThirdMapping\left(\alpha, v\in \overline{0,n-1} : (y) = (\beta^v), k \in \overline{0, w-2} : s = \beta^k\right)$

1. $i \leftarrow v+k$;

2. $\varphi^{-1}\left((y),s\right) \leftarrow InverseFirstMapping\left(\alpha, i\right)$;

3. $return\ \varphi^{-1}\left((y),s\right)$.

Лемма 12. Алгоритм 13 построения отображения φ^{-1} имеет временную сложность $O\left((\log_2 w^{m+1})^{1+\log_2 3}\right)$.

Доказательство. Шаг 1 выполняется за время $O\left(\log_2 w^{m+1}\right)$. Согласно лемме 11 шаг 2 выполняется за время $O\left((\log_2 p)^{1+\log_2 3}\right)$. Итак, временная сложность алгоритма равна $O\left(\log_2 w^{m+1} + (\log_2 p)^{1+\log_2 3}\right) = O\left((\log_2 w^{m+1})^{1+\log_2 3}\right)$, так как $p < w^{m+1}$.

Опишем процедуру восстановления секрета $x \in Z^*_p$, в которой участвуют $t \in \overline{2,m}$ проверяющих. Рассмотрим $\varphi^{-1}\left((x), s = \beta^k\right)$, где $(x)\in PG(m,w)$ — секретная точка, $s = \beta^k \in F^*_w$, $k \in \overline{0, w-2}$.

В [1] установлено, что $s \in F^*_w$ является частью публичного ключа, так же, как и проективная прямая $l \subset PG(m,w)$, содержащая секретную точку. Напомним, что T — множество секретных долей.

Алгоритм 14. Восстановление секрета.

$RevealSecret\left(VS^t, \alpha : Z^*_p =<\alpha>, \beta : F^*_{w^{m+1}} =<\beta>, w, t, T, l, k \in \overline{0, w-2} : s = \beta^k\right)$

1. $M \leftarrow BasisSubspaceConstruction\left(VS^t, \beta, w, t, T\right)$;

2. $(x) \leftarrow M \cap l$;

3. $x \leftarrow InverseThirdMapping\left(\alpha, v \in \overline{0, (w^{m+1}-1)/(w-1)-1} : (x) = (\beta^v), k\right)$;

4. $return\ x$.

Теорема 2. Временная сложность алгоритма восстановления секрета равна $O(tw^{2t})\cdot ME(w^{m+1})$.

Доказательство. Согласно лемме 10 шаг 1 выполняется за время $O(tw^{2t})\cdot ME(w^{m+1})$. Поскольку M является проективным подпространством $PG(m,w)$ размерности $t-1$, то оно содержит $\frac{w^t-1}{w-1}$ точек. Проективная прямая $l \subset PG(m,w)$ содержит $w+1$ точек. Итак, действуя методом полного перебора, можно найти пересечение на шаге 2 за время $O\left(\left(\frac{w^t-1}{w-1}\right)^2\right) = O\left(w^{2t}\right)$. Согласно лемме 12 шаг 3 выполняется за время $O\left((\log_2 w^{m+1})^{1+\log_2 3}\right)$. Таким образом, временная сложность алгоритма равна

$$O(tw^{2t})\cdot ME(w^{m+1})+O\left(w^{2t}\right)+O\left((\log_2 w^{m+1})^{1+\log_2 3}\right)=O(tw^{2t})\cdot ME(w^{m+1}).$$

Выводы. В данной работе построены формальные алгоритмы, необходимые для реализации метода порогового разделения секрета, применяемого для проведения электронного голосования. Также приведена временная сложность построенных методов. Наглядно продемонстрировано, что описанная криптосистема представляет собой полиномиальный детерминированный алгоритм. Подходящим выбором начальных параметров системы будут задание большой размерности проективного пространства и конечного поля, над которым оно задано, а также порядка мультипликативной группы, где лежит секрет. При таком выборе начальных параметров можно утверждать, что разработанные криптографические методы надежны, т. е. их применение гарантирует с высокой вероятностью проведение честных и независимых электронных выборов.

Библиографический список

1. Архангельская, Н. С. Математическая модель электронного голосования на основе методов пороговой криптографии [Электронный ресурс] / Н. С. Архангельская, А. В. Мазуренко // Системный анализ, управление и обработка информации : сб. тр. VI междунар. семинара. — Ростов-на-Дону, 2015. — Т. 1. — С. 275–280. — Режим доступа: http://ntb.donstu.ru/content/2015421/ (дата обращения: 16.10.16).

2. ElGamal, T. A public-key cryptosystem and a signature scheme based on discrete logarithms / T. ElGamal // IEEE Transactions on Information Theory. — 1985. — Vol. 31, № 4. — P. 469–472.

3. Основы криптографии / А. П. Алферов [и др.]. — Москва : Гелиос-АРВ, 2001. — 480 с.

4. A heuristic quasi-polynomial algorithm for discrete logarithm in finite fields of small characteristic / R. Barbulescu [et al.] // Advances in Cryptology — EUROCRYPT 2014 : Annual International Conference on the Theory and Applications of Cryptographic Techniques. — 2014. — Vol. 8441. — P. 1–16.

5. Stallings, W. Computer security: principles and practice / W. Stallings. — Boston : Pearson, 2012. — 182 p.

6. Рябко, Б. Я. Криптографические методы защиты информации / Б. Я. Рябко, А. Н. Фионов. — Москва : Горячая линия — Телеком, 2005. — 229 с.

7. Joux, A. The past, evolving present and future of discrete logarithm / A. Joux, A.-M. Odlyzko, C. Pierrot // Open Problems in Mathematics and Computational Science. — Cham : Springer, 2014. — P. 5–36.

8. Могилевская, Н. С. Пороговое разделение файлов на основе битовых масок: идея и возможное применение / Н. С. Могилевская, Р. В. Кульбикаян, Л. А. Журавлев // Вестник Дон. гос. техн. ун-та. — 2011. — Т. 11, № 10. — С. 1749–1755.

9. Fast Integer Multiplication Using Modular Arithmetic / De Anindya [et al.] // SIAM Journal on Computing. — 2013. — Vol. 42, № 2. — P. 685–699.

10. Schneier, B. Applied Cryptography: Protocols, Algorithms, and Source Code in C / B. Schneier. — 2nd Edition. — New York : John Wiley & Sons, 1995. — 792 p.

References

1. Arkhangelskaya, N.S., Mazurenko, A.V. Matematicheskaya model' elektronnogo golosovaniya na osnove metodov porogovoy kriptografii. [Mathematical model of electronic voting based on threshold cryptography methods.] Sistemnyy analiz, upravlenie i obrabotka informatsii: sb. tr. VI mezhdunar. seminara. [System analysis, control and information processing: Proc. VI Int. Workshop.] Rostov-on-Don, 2015, vol. 1, pp. 275–280. Available at: http://ntb.donstu.ru/content/2015421/ (accessed: 16.10.16) (in Russian).

2. ElGamal, T. A public-key cryptosystem and a signature scheme based on discrete logarithms. IEEE Transactions on Information Theory, 1985, vol. 31, no. 4, pp. 469–472.

3. Alferov, A.P., et al. Osnovy kriptografii. [Cryptography fundamentals.] Moscow: Gelios-ARV, 2001, 480 p. (in Russian).

4. Barbulescu, R., et al. A heuristic quasi-polynomial algorithm for discrete logarithm in finite fields of small characteristic. Advances in Cryptology — EUROCRYPT 2014: Annual International Conference on the Theory and Applications of Cryptographic Techniques. 2014, vol. 8441, pp. 1–16.

5. Stallings, W. Computer security: principles and practice. Boston: Pearson, 2012, 182 p.

6. Ryabko, B.Y., Fionov, A.N. Kriptograficheskie metody zashchity informatsii. [Cryptographic methods of information security.] Moscow: Hot Line — Telecom, 2005, 229 p. (in Russian).

7. Joux, A., Odlyzko, A.-M., Pierrot, C. The past, evolving present and future of discrete logarithm. Open Problems in Mathematics and Computational Science. Cham: Springer, 2014, pp. 5–36.

8. Mogilevskaya, N.S., Kulbikayan, R.V., Zhuravlev, L.A. Porogovoe razdelenie faylov na osnove bitovykh masok: ideya i vozmozhnoe primenenie. [Threshold separation of files based on bit masks: idea and potential application.] Vestnik of DSTU, 2011, vol. 11, no. 10, pp. 1749–1755 (in Russian).

9. De Anindya, et al. Fast Integer Multiplication Using Modular Arithmetic. SIAM Journal on Computing, 2013, vol. 42, no. 2, pp. 685–699.

10. Schneier, B. Applied Cryptography: Protocols, Algorithms, and Source Code in C. 2nd ed. New York: John Wiley & Sons, 1995, 792 p.

Поступила в редакцию 22.03.2017
Сдана в редакцию 29.03.2017
Запланирована в номер 17.07.2017

Received 22.03.2017
Submitted 29.03.2017
Scheduled in the issue 17.07.2017

Об авторах:

Черкесова Лариса Владимировна,
профессор кафедр «Математика и информатика» и «Кибербезопасность информационных систем» Донского государственного технического университета (РФ, 344000, г Ростов-на-Дону, пл.Гагарина,1), доктор физико-математических наук, кандидат технических наук, доцент,
ORCID: http://orcid.org/0000-0002-9392-3140
chia2002@inbox.ru

Сафарьян Ольга Александровна,
старший преподаватель кафедры «Кибербезопасность информационных систем» Донского государственного технического университета (РФ, 344000, г Ростов-на-Дону, пл.Гагарина,1), кандидат технических наук, доцент,
ORCID: http://orcid.org/0000-0002-7508-913X
safari_2006@mail.ru

Мазуренко Александр Вадимович,
студент кафедры «Кибербезопасность информационных систем» Донского государственного технического университета (РФ, 344000, г. Ростов-на-Дону, пл. Гагарина, 1),
ORCID: http://orcid.org/0000-0001-9541-3374
mazurencoal@gmail.com

Архангельская Надежда Сергеевна,
студентка кафедры «Кибербезопасность информационных систем» Донского государственного технического университета (РФ, 344000, г. Ростов-на-Дону, пл. Гагарина, 1),
ORCID: http://orcid.org/0000-0003-1678-2038
arh.iv@bk.ru

Authors:

Cherkesova, Larisa V.,
professor of the Mathematics and Computer Sciences, and Cybersecurity of IT Systems Departments, Don State Technical University (RF, 344000, Rostov-on-Don, Gagarin sq., 1), Dr.Sci. (Phys.-Math.), Cand.Sci. (Eng.), associate professor,
ORCID: http://orcid.org/0000-0002-9392-3140
chia2002@inbox.ru

Safaryan, Olga A.,
senior lecturer of the Cybersecurity of IT Systems Department, Don State Technical University (RF, 344000, Rostov-on-Don, Gagarin sq., 1), Cand.Sci. (Eng.), associate professor,
ORCID: http://orcid.org/0000-0002-7508-913X
safari_2006@mail.ru

Mazurenko, Alexander V.,
student of the Cybersecurity of IT Systems Department, Don State Technical University (RF, 344000, Rostov-on-Don, Gagarin sq., 1),
ORCID: http://orcid.org/0000-0001-9541-3374
mazurencoal@gmail.com

Arhangelskaya, Nadezhda S.,
student of the Cybersecurity of IT Systems Department, Don State Technical University (RF, 344000, Rostov-on-Don, Gagarin sq., 1),
ORCID: http://orcid.org/0000-0003-1678-2038
arh.iv@bk.ru